The 1998 OGI-FONIX Broadcast News Transcription System

Xintian Wu, Chaojun Liu, Yonghong Yan,
Doughwa Kim, Seth Cameron (FONIX), Randy Parr (FONIX)

Oregon Graduate Institute of Science and Technology,
20000 N.W. Walker Road, P.O. Box 91000,
Portland, OR 97291-1000, USA, (xintian@cse.ogi.edu)

ABSTRACT

This paper describes the OGI-FONIX large vocabulary system developed for the 1998 broadcast news evaluation. The main differences from our last year's system [1] are: (1) A multiple pass decoder is used. (2) Long periods of silence are deleted in both training and decoding features. Cepstral mean subtraction is applied individually to automatically derived speaker clusters. These methods result in a 3% absolute improvement on a self-constructed 1000 seconds test set. (3) Different dictionaries are used for the within word and crossword decoding passes. Long segments from the first pass are chopped at hypothesized word boundaries and then merged back in the crossword decoding pass. These two methods yield 2.3% absolute improvement on the hub4e_97 evaluation set. (4) The system vocabulary size has been increased from 25K to 56K. With these changes, our 1998 system achieves a word error rate 27.9% on the h4e_98_1 test set, and 23.6% on the h4e_98_2 test set for the hub4e_98 evaluation.

1. Introduction

In recent years, large vocabulary speech recognition systems have been challenged with increasingly difficult tasks. The current HUB4 DARPA competition involves transcribing audio taken directly from broadcast news programs. The challenge is to perform accurate transcription of audio that includes for example, multiple speakers, background music, and casual dialog.

The OGI LVCSR group first participated in the DARPA evaluation program in 1997. We developed our 1998 HUB4 system with support from FONIX corporation. Our 1997 system was substantially behind the rest of the competition, so our strategy for 1998 was to:

(1) Better understand the pitfalls present in broadcast news audio

(2) Study existing techniques and incorporate them into our system

(3) Develop new ideas to improve system performance.

In the following sections, we first present an overview of the system and then detail each system component. Some comparative experiments are presented that show how various modifications improved system performance.

2. System Overview

The OGI-FONIX large vocabulary speech recognition system is a continuous HMM-based system. It uses 39 MFCC feature parameters (12 mfcc parameters plus energy and their first and second derivatives). It has a 56k-word active vocabulary. The standard Hub4 acoustic training data for 1996 and 1997 released by LDC were used. Bigram and trigram language models were trained with WSJ and BN LM data released last year and this year. The system has two sets of acoustic models: a within-word triphone model for the first decoding pass, and a crossword triphone model for the second decoding pass. During multiple pass decoding, unsupervised MLLR was used to adapt the model parameters to each speaker. Monophone recognition and Bayesian Information Criterion (BIC) were used to segment/cluster the test data.

2.1. Segmentation and Clustering

Segmentation and clustering were based on the BIC method used by Chen and Gopalakrishnan from IBM [2]. The general idea behind the BIC method is to ask whetheher data collected from two hypothesized audio segments is better modeled with a single full-covariance guassian or two full-covariance gaussians. Because two gaussians use more parameters to fit the same data, their score is penalized when compared to the single gaussian fit. The BIC measure is first used to segment the audio file by placing markers between sections with sufficiently different 1st and 2nd order statistics. These segments are then clustered by combining segment pairs with the lowest mutual BIC score until a stop criterion is met.

For the h4e_98_1 test set, 321 segments were detected. Among them, 79 segments were classified as pure music and discarded.

For the h4e_98_2 test set, 422 segments were detected. Among them, 37 segments were classified as pure music and discarded.

The average length of the decoding segments was 16.46 seconds (max 125 seconds).

These segements are subsequently processed by a monophone decoder. Silence is detected and treated as a sentence boundary where long segments may be chopped into shorter ones. The average segment length after this step was 6.26 seconds (from 1 second to 16 seconds).

For memory and speed considerations, it is desirable not to process segments longer than 10 seconds with the full decoder. Therefore, a fast within-word recognizer is applied next so that segments longer than 10 seconds can be re-cut at likely word boundaries. The resulting segments had an average length of 4.8 seconds (from 1 second to 10 seconds).

2.2. Acoustic Modeling

Standard HUB4 acoustic traning data for 1996 and 1997 were used to train the acoustic models for the final system.

For the 1996 training data, overlapped segments were discarded. A force alignment program was used to verify the utterances. Those

bad-conditioned segments were rejected. About 64 hours of data remained.

Only 31 out of the 72 hours of 1997 training data survived an intensionally restricted (with a narrow pruning-beam) alignment process. It is suspected that the 1997 transcripts are less accurate than the 1996 transcripts.

During training, the 1996 training data were used to train the acoustic models (monophone and context dependent triphone) before state clustering. Next, a decision tree based state clustering algorithm was used to produce the context clustering. The triphone models were refined through several iterations of gaussian splitting and reestimation. 1997 training data were added after each set of models had at least 8 gaussians per state.

Two sets of triphone model were trained:

(1) a within-word model with 6800 distinct states (16 gaussians per state)

(2) a crossword model with 7500 distinct states (24 gaussions per state)

2.3. Language Model and Lexicon

The CMU-Cambridge language model package V2.0 was used. The text materials included the WSJ LM data and BN LM data obtained from LDC. The Good-Turing method was used to estimate back-off trigram and bigram language models. They contained 14M trigrams and 7M bigrams respectively. The perplexity of the trigram LM on last year's evaluation data was 168.

This year we extended our vocabulary from 25k to 56k based on word frequency in the training corpus. The pronunciations of these words were extracted from the fonix dictionary. Two dictionaries were used. They have the same vocabulary but different number of alternative pronunciations. The smaller dictionary (63k entries) is used by the within-word decoder, and the big dictionary (66k entries) is used by the crossword decoder.

2.4. Decoding

Last year, because a single pass decoder performed all the recognition tasks, computer memory constraints forced the use of a tight beam for state pruning. This year a multiple pass decoding strategy was adopted. The general idea follows SRI's progressive search technique [3], modified word graph generation, and extension method [4].

A within-word decoder using a bigram LM is used in the first decoding pass to generate a word graph from recorded trace-back information. Next, the word graph is extended to include crossword triphones and a trigram LM. Then, a graph decoder performs a second decoding pass on the extended word graph to generate the final transcription output. There are some modifications in our system:

(1) The two passes use different dictionaries. The two dictionaries have the same vocabulary size but different numbers of alternative pronunciations. The small dictionary is for the within-word decoding pass, and the big dictionary is for the crossword decoding pass. The purpose for using an expanded dictionary on the second pass is to better rescore alternative pronunciations of words hypothesized by the first decoding pass. See section 3 for detailed explanation.

(2) Word graphs from the same sentence are merged before the second decoding pass. Merging two word graphs means making a full connection between the words at the end of the first word graph and the words at the beginning of the second word graph. This is a reverse-process of the last step in segmentation, where long segments are chopped into shorter ones at the word boundary. Since the language model is trained on sentence base, but segments presented to the decoder can be just part of a sentence, this recombination step makes the decoding environment match more closely with the language model training environment. See section 3 for detailed explanation.

2.5. Adpatation

MLLR speaker adaptation is performed on each decoding pass. A regression tree is created with 8 leaves and 15 regression matrices. Occupation count determines which level of the tree is used for adaptation. The crossword decoding output serves as the target for within-word speaker adaptation.

3. Approaches and Experiments

In addition to the described training and decoding strategies, many attempts were made to improve system performance.

To speed up experimental turn-around time, a 1000-second subset of the 1996 PE test set data was used. This subset has the same condition (F0-FX) distribution, the same language model complexity, and the same OOV rate as the hub4e_97 test set. Most of the training and decoding experiments were based on this subset. The Hub4e_97 evaluation set was also used as a development set.

3.1. Silence Deletion and Cluster Based CMS

Cepstral Mean Subtraction (CMS) is a simple but effective front-end technique that helps normalize speaker and channel variation. Two factors affect the performance of CMS:

(1) Short segments contain insufficient data to calculate cepstral means that characterize speaker or channel variation accurately. Therefore, we attempt to cluster segments from the same source before calculating cepstral means.

(2) The cepstral mean is influenced by both speaker and channel variation. The cepstral mean of segments with long silence will reflect more channel information than speaker information. Therefore, silence is removed from segments during both training and decoding to keep the ratio of channel to speech influence relatively stable. This silence deletion technique was introduced by Yan in his work on language identification [5]. The steps to perform CMS with silence deletion are as follows:

(a) Silence detection. This is done at the segmentation stage by a monophone decoder.

(b) If silence is at the beginning of a segment, the segement start-frame is adjusted to preserve 7 frames of silence at the beginning.

(c) If silence is at the end of a segment, the segment end-frame is adjusted to preserve 7 frames of silence at the end.

(d) If silence is in the middle of a segment, the segment is shortened to preserve 8 frames of silence in the middle.

(e) Calculate CMS for all the segments.

Some experiments were run on our 1000-second development set. Word error rate results are shown in Table 1. Silence deletion yielded 2.8% improvement and cluster based CMS yielded 0.3% improvement. The mixed effect of silence deletion and cluster based CMS yielded 3% performance improvement on the subset.

1000s subset	WER
Baseline	38%
Silence Deletion	35.2%
Clustered CMS	37.7%
Both	35%

Table 1: Silence Deletion and Cluster Based CMS

3.2. Two Dictionaries

As mentioned above, two dictionaries were used in the final decoding system. They have the same vocabulary size (56K) but different numbers of alternative pronunciations (63K and 66K). The procedure to change the dictionary is illustrated in Figure 1:

for each word graph {

for each word in the word graph {

(1) Remove the alternative pronunciation marker.

(2) Find all the alternative pronunciations in the big dictionary and insert them by duplicating the corresponding arcs.

}

}

The reasons for incorporating two dictionaries in the system are:

(1) For the within-word decoding pass, computational complexity restricts the use of a very big dictionary. Also, this pass uses a bigram language model which lacks the accuracy necessary to handle more alternative pronunciations.

(2) For the crossword decoding pass, each word graph has a limited vocabulary size, so the computational complexity caused by enlarging the dictionary and incorporating a trigram language model is manageable. Including additional alternative pronunciations during the crossword decoding pass increases acoustic probability scores.

Experiments were done on the HUB4e_97 evaluation set. The word error rate results are shown in Table 2. In the third row of the table, without segment merging, the use of two dictionaries had 0.2% improvement. In the fourth row of the table, with segment merging, the use of two dictionaries had 0.3% improvement.

3.3. Chop and Merge Segment

In the segmentation stage, long segments are chopped first at the sentence boundary (silence detected by the monophone decoder) and then at the word boundary (detected by the fast word recognizer). The resulting segments can be any part of a sentence which introduces a mismatch between the training and decoding environment since the language model is trained on sentence base. Mistaken sentence boundary tokens likely reduce system performance.

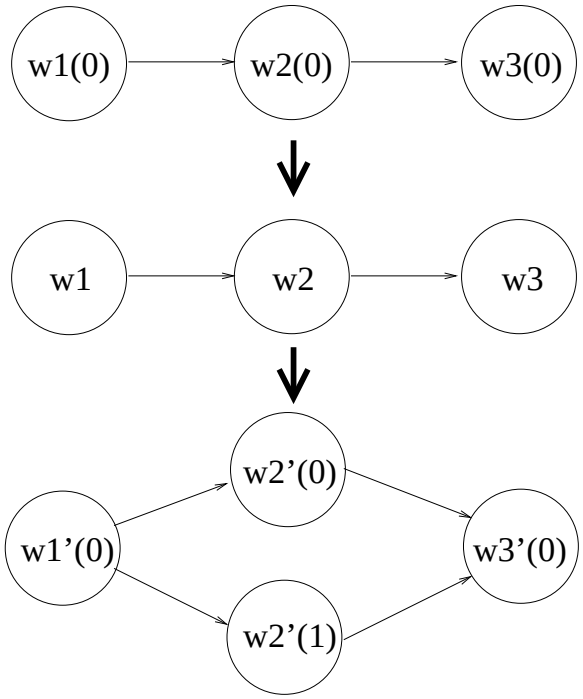

Figure 1: Procedure to Change Dictionary
w1, w2, and w3 are recognized words in the small dictionary. w1', w2', and w3' are corresponding words in the big dictionary. "(0)" and "(1)" are alternative pronunciation markers. In the figure it is assumed that word w2 has two alternative pronunciations in the big dictionary, noted as w2'(0) and w2'(1).

CMU [6] attempted to address this mismatch by "introducing two words of context and re-training the language model." The conclusion was that "the standard technique of modeling the begin-of-sentence token and assuming the end-of-sentence token provided the lowest word error rate."

We addressed the problem in a different way. In our 1998 system, we merge back segments chopped from the same long utterance to form complete sentences before the crossword decoding pass. Thus, the boundary issue caused by segmentation is alleviated without requiring that we modify our language model.

The procedure to merge two segments (word graphs) is illustrated

hub4e_97 WER	1 dict only	using two dicts
Long segments (<30s)	30.0%	-
chop segment (<10s)	29.3%	29.1%
chop and merge segment	28.0%	27.7%

Table 2: Using two dictionaries in recognition

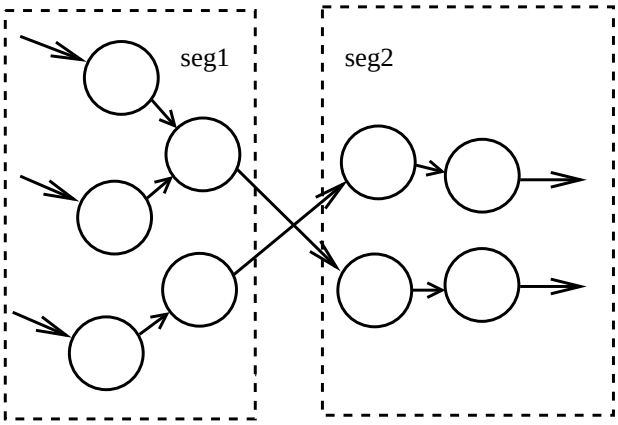

Figure 2: Procedure to merge two segments
Segments from the same sentence are merged back to form a complete sentence. The resulting word graph is recognized by a crossword decoding pass.

in Figure 2. The words at the end of the first word graph and the words at the beginning of the second word graph are fully connected. The temporal order of the segments is recorded to ensure correct merging.

Experiments were done on the HUB4e_97 test set. The word error rate results are shown in Table 2. In the first column of the table, chopping long segments into shorter ones yielded 0.7% improvement (shorter segments allow the within-word pruning beam to be relaxed). Merging the small segments back yielded another 1.3% improvement. The second column shows that the combined effects of using two dictionaries, chopping, and merging provided a 2.3% improvement.

References

1. Yonghong Yan, Xintian Wu, Johan Schalkwyk, and Ron Cole, “Development of CSLU LVCSR: The 1997 DARPA Hub4 EVALUATION System”, Broadcast News Transcription and Understand Workshop, 1998.
2. S. S. Chen and P. S. Gopalakrishnan, “Clustering via the Bayesian Information Criterion with applications in speech recognition”, Proc. ICASSP, vol. 2, page 645 648, 1998.
3. Hy Murveit, John Butzberger, Vassilios Digalakis and Mitch Weintraub, “Large Vocabulary Dictation Using SRI’s Decipher Speech Recognition System: Progressive search Techniques”, Proc. ICASSP, vol. 2, page 319 322, 1993.
4. Annath Sankar, Fuliang Weng, Ze’ev Rivlin, Andreas Stolcke and Ramana Rao Gadde, “The Development Of SRI’s 1997 Broadcast News Transcription System”, Broadcast News Transcriiption and Understand Workshop, 1998.
5. Yonghong Yan, “Development Of An Approach To Language Identification Based On Language-Dependent Phone Recognition”, PhD Thesis, Oregon Graduate Institute, 1995.
6. K. Seymore, S. Cheng, S. Doh, M. Eskenazi, E. Gouvea, B. Raj, M. Ravishankar, R. Rosenfeld, M. Siegler, R. Sterm and E. Thayer, “The 1997 CMU Sphinx-3 English Broadcast News Transcription System”, Broadcast News Transcription and Understand Workshop, 1998.